Neural Orientation- and Object-Representations for Robust and Reliable Object Detection on Aerial Image Datasets

Kai Konen
Institute for Software Technology
German Aerospace Center (DLR)
Cologne, Germany
kai.konen@dlr.de

Abstract

Machine learning-based models for object detection heavily rely on large datasets of labeled images. When models trained on these datasets are applied to Unmanned Aerial Vehicle (UAV) imagery, the problem arises that the conditions under which the training images were created (lighting, altitude, angle) may be different to the UAVs applied conditions, leading to misclassifications. This problem becomes even more pressing in safety critical applications where failures can have huge negative impacts and constitute obstacles for certification of cognitive UAV components. In earlier work, we demonstrated that synthetic images can be used to evaluate neural networks trained on real-world data and that artificial images, under certain circumstances, can have a positive effect on object detection performance. Our findings suggest that photo-realism is not as important as a well-balanced dataset in terms of altitude and object-orientation distribution. Based on these findings we train convolutional neural networks on synthetic images with orientation- and object-class-annotations. At inference time we record and visualize outputs of the activation-functions of different hidden-layers for all training- and test-images. These neural representations allow for a better understanding of how and what the neural network has learned during training. We demonstrate this by comparing the neural representations of networks trained on discontinuous- and continuous rotation representations.

Index Terms

Regression, Image Classification, SO(3), SO(2), Rotation Matrix, Rotation Representations, UAVs

I. Introduction

Unmanned Aerial Vehicles (UAVs) equipped with cameras and intelligent components for computer vision are a promising mean in various applications for smart cities, traffic surveillance, rescue, and many others. In this context, an important task is object detection and recognition in aerial imagery, which is supported by recent advances in machine learning, especially deep learning.
Since UAVs are safety critical systems, there are very strict requirements concerning reliability and transparency of actions resulting from decisions made by artificial intelligence. This concerns robustness of models against adversarial inputs, early identification of errors and biases, as well as transparency and controllability.

A. Problem Statement

While verification techniques and improved training methods can guarantee model robustness against adversarial inputs to a certain extent, some uncertainties remain when these models are put into operation. Even if models are trained and verified on very large datasets, it is not given that in real situations there can be inputs that are very different from the data used for training, which can lead to unexpected and potentially unintended behavior. Among others, reasons can be changes in the operation environment during the time when training-data was collected and the time when the model is applied. This is especially the case when pre-trained models based on open datasets are used in real applications where the creation of training-data is costly and often impractical. For example, if the recorded images show objects only in certain orientations, it is not given that the system works well for orientations not represented in the training- or test-data.

B. Research Objective

Developing new object detection methods on images is often done by using large datasets such as COCO [1], KITTI [2] or VisDrone [3] for training and validation. Although object detection methods can perform well on the source domain they were trained and validated on, performance often drops when put into operation in a target-domain with a shifted data distribution. The reason is that certain features of the target-domain may differ from inherent domain properties of the source domain [4]. For aerial images these distribution shifts may be changes in weather, time of day, lighting, altitude of the aerial vehicle, object-orientation or the environment where the images are taken. If a model happens to fail in practice because the real situations deviate too much from those represented in the training-data, there are two options to mitigate this problem (a)

assembling and labelling a new dataset, which is costly and sometimes impractical, or (b) analyzing the target-domain and augmenting the training-dataset or the detection model in one way or another.
Building on previous work [5] we evaluate how the relative rotation of an object towards a camera can be learned from rgb-images and how this information can be used to increase image classification performance. Instead of using aerial imagery or UAV datasets, we use a dataset especially created for learning rotation representations. Furthermore we compare the difference in neural network layer activity and loss-functions during learning for discontinuous and continuous orientation representations.

II. Related work

In recent years more and more research has been done to improve object detection on aerial and UAV image datasets [6] [7]. Nevertheless most of the approaches focus on improving object detection on single datasets or on high altitude aerial images and do not take the transferability of trained models between datasets or the use of synthetic data into account, which is the research focus of this work.
As shown by Tremblay et al. [8] as well as Nowruzi et al. [9] object detection neural networks trained on synthetic data can achieve considerable accuracy on real-world images. Both focus on improving object detection for autonomous driving tasks, where the camera angles, height and image content highly differ from our object detection task on aerial imagery.
Bondi et al. have used the AirSim simulator [10] to generate simulated infrared images of the African savanna for wildlife conservation with UAVs and noted that the creation of synthetic datasets can save costs and labour when compared to recording and labelling real datasets. They have created a simulated environment based on the African savanna and compared the performance of object detection models trained on real infrared images with models trained on their simulated data [11]. They found that models trained only on simulated data produced the best recall results for small animals and models trained only on real data the best precision results. Bondi et al. also released a dataset consisting of real and synthetic labelled aerial infrared videos of the African savanna [12] to develop and evaluate object detection, tracking as well as domain adaptation methods for aerial infrared images.
Apart from object detection on RGB and infrared images, research is also being done for object detection and instance segmentation on LiDAR data recorded from UAVs [13] [14], though available datasets are even more scarce than datasets based on RGB images. Another application for simulated UAV environments was developed by Sadeghi and Levine [15]. They trained a reinforcement learning neural network in a simulated environment and were able to transfer the learned behavior to a real UAV, demonstrating additional use cases of UAV simulations. A lot of work has been done in the field of domain adaptation [16]. The approaches range from supervised domain adaption [17], generative adversarial approaches [18] to unsupervised approaches [19]. Similar to Kiefer, Messmer and Zell [20] we design a multi-head architecture but instead of the discrete binning of altitude and camera orientation, we designed an object classification head and a regression head than regresses continuous and discontinuous rotation representations. Sundermeyer et al. [21] proposed a pipeline for 6d object detection on rgb-images based on autoencoders code-books. While they use a code-book approach to estimate the objects' rotation matrix, we try to regress the rotation matrix directly and visualize continuous and discontinuous rotation representations.

III. Methods

In the following section we will describe which neural network architecture we chose and how we performed the dimensionality reduction of the hidden-layer activation-functions. Further, we elaborate why we use continuous and discontinuous representations for rotations in $\mathbb{R}^2$ and $\mathbb{R}^3$, as well as how we generated the dataset used in our experiments. Albeit it is possible to use discrete as well as discontinuous rotation representations, we postulate that object rotations in the real-world are continuous. Therefore it should be harder for a neural network to fit to a discontinuous or even discrete rotation representation [22].

A. Rotation Representation

There are various ways to describe rotations in $3D$ space. The most commonly used representations are Euler angles, quaternions [23] and the rotation matrix from the $SO(3)$ group. As described by Zhou et al. [22] Euler angles and quaternions are discontinuous and therefore difficult for neural networks to learn. For Euler angles this discontinuation lies between 2π and 0 radians, or 360 and 0 degree (see Fig. 1). The 3×3 $SO(3)$ rotation matrix and its $6D$ version proposed by Zhou et al. [22] do not have this discontinuation. For rotations around a single axis, as suggested by Zhou et al. [22], we use the first column vector of the $SO(2)$ rotation group $[\cos\theta, \sin\theta]$ for the continuous representation and Euler angles as the discontinuous representation.

B. Neural Network Architecture

For the backbone of our neural network models we use the convolutional layers of the deep residual neural network ResNet-18 [24]. We replace the original classification head of the ResNet-18 model with a fully connected 512×256 linear layer. On top of that we add a regression and a classification head. As the optimization algorithm we use Adam [25]. For the regression

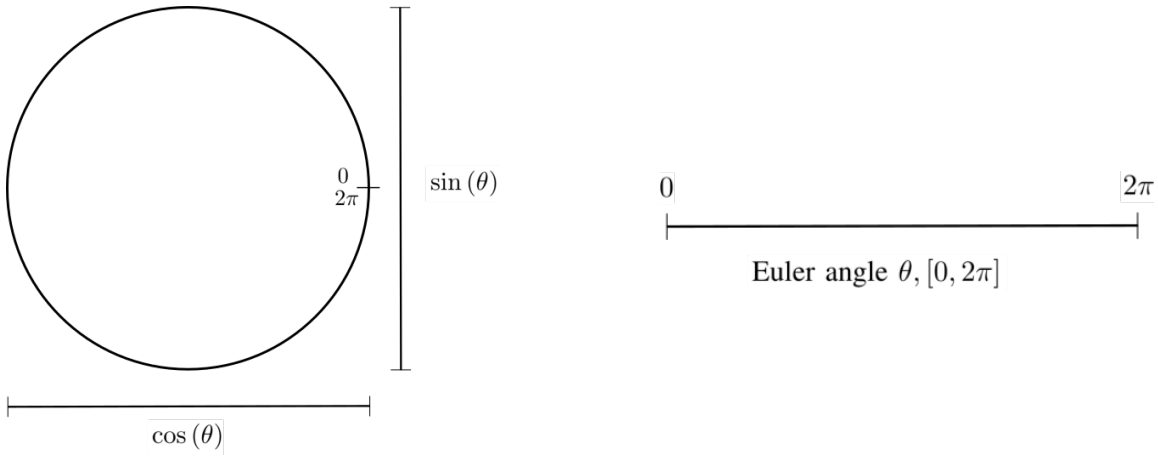


Fig. 1: Continuous and discontinuous representations of a rotation around a single axis.

loss function of the neural network we chose the squared-L2 norm (MSE-Loss) and for the classification loss function we use the Cross-Entropy-Loss. The output of the regression head consists of two linear outputs for rotations around a single axis and six linear outputs for rotations around all three axes for the continuous representations. For the discontinuous representation we used a single output for the regression head. The output of the classification head consists of 33 linear outputs, one for each class of our dataset. The neural networks takes rgb-images of objects as an input and tries to regress the relative rotation matrices of each object depicted on the rgb-image, as well as the corresponding object class.

C. Dimensionality Reduction

As described above, rotations around a single axis can be continuously described with two parameters. Since we try to get a better understanding of how the neural network learns, we apply a principal component analysis [26] on the outputs of the activation-functions of different hidden-layers in order to transform the data from the unordered hidden feature-space into a coordinate system where the greatest variance of the data lies on the first component, the second greatest variance on the second and so on. The first n components are therefore the components that best fit the observed data. The first two components should therefore describe rotations in around a single axis while the first six components describe rotations around three axis. To keep the visualizations intuitive we will focus on $\mathbb{R}^2$ rotations.

D. Dataset

Fig. 2: Example images of two different objects from the generated dataset.

Since most UAV datasets do not annotate the camera orientation, and if only in low resolution bins, we decided to use the $3D$ models from the HomebrewedDB dataset [27]. We used Blenderproc [28] to render the objects and generate the annotations. We rendered all objects multiple times while rotating the object around its z-axis at an angle increment of 0.36 degree. We recorded each rendered rgb-image as well as the relative rotation matrix. Overall we generated $33,000$ images depicting 33 different objects. See Fig.2 for examples of the generated data.

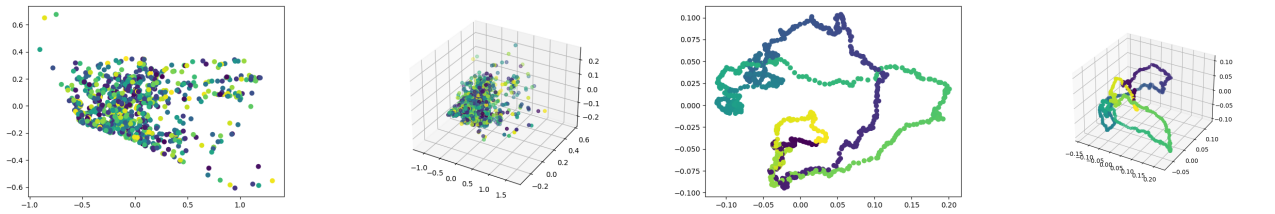
IV. Experiments

As described in Section III-B, we trained a multi-head model. One head on rotation annotations in order to learn object rotations from rgb-images and one head on the class annotations to test whether the rotation information can be used to improve object classification performance. We ran multiple training runs with different setups: Training only the rotation head, only the classification head, training both head simultaneously and lastly training first the rotation head and then the classification head. We trained the models on both, continuous and discontinuous representations. During training, we saved the neural network weights every 10 epochs. At inference time, we recorded the output of the activation-functions of the fully connected hidden-layers. We then performed a principal component analysis on various splits of the recorded activations.

V. Results

Training both heads simultaneously did not lead to better classification performance, as both heads converged at higher error-rates compared to training each head individually. Training the regression head first and then the classification head did also not lead to improved classification results.

Interestingly, as seen in Fig. 3, even with completely random weights, one can clearly see the difference between the activations of random sampled images and images belonging to a single object. During training, we can observe how the activations of the hidden layer slowly get untangled until a representation is found which fits the regression problem. This can be observed for both, continuous and discontinuous representations (see Fig. 4). However as seen in Fig. 4h the neural network has problems separating rotations around 2π and 0 radians when fitting to the discontinuous representation.



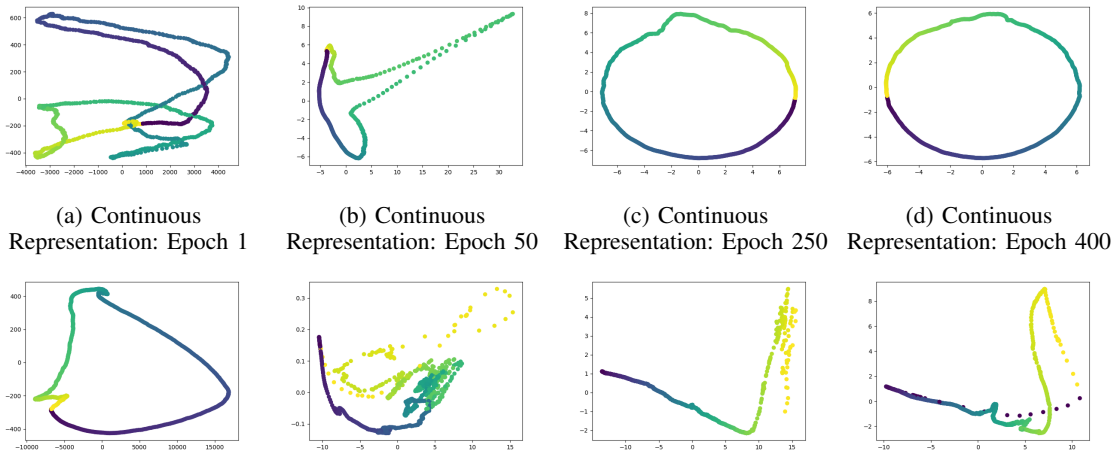
(a) Activation of 1000 random sampled images.

(b) Activation of 1000 random sampled images (3D).

(c) Activation of 1000 images of one object.

(d) Activation of 1000 images of one object (3D).

Fig. 3: Plot of the first two and three principal components from the activation of the fully connected hidden layer *before* training. All weights are random initialized. Coloring in (a) and (b) is random. Coloring in (c) and (d) is a color-scale from 0 radians (yellow) to 2π radians (purple).



(a) Continuous Representation: Epoch 1

(b) Continuous Representation: Epoch 50

(c) Continuous Representation: Epoch 250

(d) Continuous Representation: Epoch 400

(e) Discontinuous Representation: Epoch 1

(f) Discontinuous Representation: Epoch 50

(g) Discontinuous Representation: Epoch 250

(h) Discontinuous Representation: Epoch 400

Fig. 4: Activations of the fully connected hidden layer at different training epochs for the continuous and discontinuous rotation representation.

VI. Discussion

We were not able to improve object classification performance by using additional information about the object relative rotation. For the case of simultaneous training, we assume this is due to the two loss functions interfering with each other during the optimization step. Nevertheless we still think that incorporating information about the object orientation can be beneficial for object classification, but not with this architecture choice and training setup. When looking at Fig. 3c, one can already see an continuous, albeit entangled, representation of the objects rotation. During training then, the neural network fits both, the continuous and discontinuous representation (compare Fig. 1 with Fig. 4c and Fig. 4g). However, the discontinuous representation is difficult for the neural network to fit, as one would expect in accordance with Zhou et al. [22]. Looking at Fig. 4h one can see how the neural network tries to group the rotations around 0 and 2π together, similar to the continuous representation, leading to a higher error for the discontinuous representation.

VII. Conclusion and future work

As seen above, continuous rotation representations are preferable over discontinuous representations. Future work includes comparing continuous rotation representations with discrete binned approaches like Kiefer, Messmer and Zell [20] for UAV imagery. Further, code-book based approaches like Sundermeyer et al. [21] could be applied to the UAV imagery domain. Lastly, more work needs to be done to directly regress continuous rotation representations and incorporate said representations in more robust object detection models.

References

[1] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and L. Zitnick, "Microsoft COCO: Common objects in context," in *ECCV*. European Conference on Computer Vision, Sep. 2014. [Online]. Available: https://www.microsoft.com/en-us/research/publication/microsoft-coco-common-objects-in-context/

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[3] P. Zhu, L. Wen, D. Du, X. Bian, Q. Hu, and H. Ling, "Vision meets drones: Past, present and future," *CoRR*, vol. abs/2001.06303, 2020. [Online]. Available: https://arxiv.org/abs/2001.06303

[4] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," 2016. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12443

[5] K. Konen and T. Hecking, "Increased robustness of object detection on aerial image datasets using simulated imagery," in *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, 2021, pp. 1–8.

[6] Y. Xu, G. Yu, Y. Wang, X. Wu, and Y. Ma, "Car detection from low-altitude UAV imagery with the faster R-CNN," *Journal of Advanced Transportation*, vol. 2017, pp. 1–10, Aug. 2017.

[7] A. Bouguettaya, H. Zarzour, A. Kechida, and A. M. Taberkit, "Vehicle detection from UAV imagery with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2021.

[8] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2018.

[9] F. E. Nowruzi, P. Kapoor, D. Kolhatkar, F. A. Hassanat, R. Laganière, and J. Rebut, "How much real data do we actually need: Analyzing object detection performance using synthetic and real data," *CoRR*, vol. abs/1907.07061, 2019. [Online]. Available: http://arxiv.org/abs/1907.07061

[10] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[11] E. Bondi, D. Dey, A. Kapoor, J. Piavis, S. Shah, F. Fang, B. Dilkina, R. Hannaford, A. Iyer, L. Joppa, and M. Tambe, "AirSim-W: A simulation environment for wildlife conservation with UAVs," in *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, ser. COMPASS '18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: https://doi.org/10.1145/3209811.3209880

[12] E. Bondi, R. Jain, P. Aggrawal, S. Anand, R. Hannaford, A. Kapoor, J. Piavis, S. Shah, L. Joppa, B. Dilkina, and M. Tambe, "BIRDSAI: A dataset for detection and tracking in aerial thermal infrared videos," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 1736–1745.

[13] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. García, and A. De La Escalera, "Birdnet: A 3d object detection framework from lidar information," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 3517–3523.

[14] Z. Ye, Y. Xu, R. Huang, X. Tong, X. Li, X. Liu, K. Luan, L. Hoegner, and U. Stilla, "Lasdu: A large-scale aerial lidar dataset for semantic labeling in dense urban areas," *ISPRS International Journal of Geo-Information*, vol. 9, no. 7, 2020. [Online]. Available: https://www.mdpi.com/2220-9964/9/7/450

[15] F. Sadeghi and S. Levine, "CAD2RL: Real Single-Image Flight without a Single Real Image," *arXiv:1611.04201 [cs]*, Jun. 2017, arXiv: 1611.04201. [Online]. Available: http://arxiv.org/abs/1611.04201

[16] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231218306684

[17] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa, "Cross-domain weakly-supervised object detection through progressive domain adaptation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5001–5009.

[18] H.-K. Hsu, W.-C. Hung, H.-Y. Tseng, C.-H. Yao, Y.-H. Tsai, M. Singh, and M.-H. Yang, "Progressive domain adaptation for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2019.

[19] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4888–4897.

[20] B. Kiefer, M. Messmer, and A. Zell, "Diminishing domain bias by leveraging domain labels in object detection on uavs," in *2021 20th International Conference on Advanced Robotics (ICAR)*, 2021, pp. 523–530.

[21] M. Sundermeyer, Z.-C. Marton, M. Durner, and R. Triebel, "Augmented autoencoders: Implicit 3d orientation learning for 6d object detection," *International Journal of Computer Vision*, vol. 128, no. 3, pp. 714–729, 2020.

[22] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5745–5753.

[23] W. R. Hamilton, "Theory of quaternions," *Proceedings of the Royal Irish Academy (1836-1869)*, vol. 3, pp. 1–16, 1844. [Online]. Available: http://www.jstor.org/stable/20489494
[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[26] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.
[27] R. Kaskman, S. Zakharov, I. Shugurov, and S. Ilic, "Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects," *International Conference on Computer Vision (ICCV) Workshops*, 2019.
[28] M. Denninger, M. Sundermeyer, D. Winkelbauer, Y. Zidan, D. Olefir, M. Elbadrawy, A. Lodhi, and H. Katam, "Blenderproc," *arXiv preprint arXiv:1911.01911*, 2019.